

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## SIMULÁTOR DLMS KONCENTRÁTORU

DLMS CONCENTRATOR SIMULATOR

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Ondřej Buš

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Lieskovan

BRNO 2020

# Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

**Student:** Ondřej Buš

**ID:** 195821

**Ročník:** 3

**Akademický rok:** 2019/20

**NÁZEV TÉMATU:**

## Simulátor DLMS koncentrátoru

### POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je vytvoření funkčního koncentrátoru pro chytré elektroměry (smart metry) komunikujícího pomocí DLMS/COSEM a TCP/IP. Softwarový koncentrátor by měl být sepsán v jazyku Java nebo Python, bude podporovat vzdálený odečet spotřebované elektrické energie a výčet bezpečnostních incidentů na jednotlivých elektroměrech. Součástí řešení by mělo být i jednoduché webové rozhraní, které bude umožňovat koncentrátor spravovat a graficky zobrazovat průběh odběru elektrické energie v čase.

### DOPORUČENÁ LITERATURA:

[1] Gurux DLMS Client: <https://www.gurux.fi/Gurux.DLMS.Client>

[2] JIRKA, Bc Matěj. Framework DLMS/COSEM pro sběr dat v AMM systémech.

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 8.6.2020

**Vedoucí práce:** Ing. Tomáš Lieskovan

**doc. Ing. Jan Hajný, Ph.D.**  
předseda rady studijního programu

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Práce se zaměřuje na problematiku chytrých elektroměrů a datových koncentrátorů. V první části práce lze nalézt úvod do DLMS/COSEM. Zde jsou vysvětleny základní principy a nástroje komunikace pomocí DLMS protokolu. Dále je tato část zaměřena na COSEM třídy a objekty, sloužící pro přístup k naměřeným datům. V poslední řadě je nastíněno odkazování se na data pomocí OBIS identifikátorů.

V části, která se zabývá samotnou realizací koncentrátoru budou uvedeny jednotlivé funkční celky koncentrátoru – procesem komunikace se zařízeními a správou konfigurací. Nakonec se práce zaměří na popis jednotlivých částí webového rozhraní umožňující správu chytrých elektroměrů na koncentrátoru – přidávání, editace a odebrání chytrých zařízení.

## KLÍČOVÁ SLOVA

COSEM, DLMS, Chytrý elektroměr, Java, koncentrátor, OBIS, PHP

## ABSTRACT

This work is focused on a field of smart metering and data concentrators. The first part of this paper describes the DLMS/COSEM specification. The first section contains DLMS protocol specification, the process of communication between devices and services that allows data communication. COSEM classes and objects can be found in the second section. These objects are used to access a specific data stored inside smart meters. Lastly, there is an section that is focused on OBIS objects which are used for addressing. In the second part, the concentrator's basic functionalities are described. These functionalities include device management and configuration file storage.

The last part then focuses on a web server. The web server allows a user to manage the concentrator by adding, editing and removing connected smart meters.

## KEYWORDS

Concentrator, COSEM, DLMS, Java, OBIS, PHP, Smart meter

BUŠ, Ondřej. *Simulátor DLMS koncentrátoru*. Brno, 2020, 61 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Tomáš Lieskovan

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Simulátor DLMS koncentrátoru“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Tomáši Lieskovanovi, za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

# Obsah

<b>Úvod</b>	<b>19</b>
<b>1 Úvod do chytrých měřicích zařízení</b>	<b>21</b>
1.1 Situace v Evropské unii . . . . .	23
1.2 Situace v České republice . . . . .	24
<b>2 DLMS/COSEM</b>	<b>25</b>
2.1 DLMS/COSEM na modelu ISO/OSI . . . . .	26
2.2 DLMS . . . . .	27
2.2.1 Systémový název zařízení . . . . .	27
2.2.2 Proces komunikace . . . . .	27
2.2.3 Specifikace služeb . . . . .	27
2.2.4 Primitivní asociační služby . . . . .	29
2.2.5 Bezpečnostní koncepty . . . . .	29
2.3 COSEM třídy rozhraní . . . . .	32
2.3.1 Struktura chytrého zařízení . . . . .	32
2.3.2 Druhy přístupu ke COSEM objektům . . . . .	33
2.3.3 Druhy COSEM tříd rozhraní . . . . .	34
2.4 OBIS . . . . .	35
2.4.1 OBIS třídy . . . . .	35
2.4.2 OBIS identifikátor . . . . .	36
<b>3 Koncentrátor chytrých elektroměrů</b>	<b>37</b>
3.1 Použité knihovny . . . . .	37
3.2 Použitá zařízení . . . . .	38
3.3 Využité technologie . . . . .	38
3.3.1 Java . . . . .	38
3.4 Odklon od specifikace DLMS/COSEM . . . . .	39
3.4.1 Vytvoření spojení . . . . .	39
3.4.2 Chybná implementace časové zóny . . . . .	40
3.5 Funkční celky koncentrátoru . . . . .	41
3.5.1 Komunikace s chytrými zařízeními . . . . .	41
3.5.2 Správa konfigurace . . . . .	43
3.5.3 Komunikace s webovým serverem . . . . .	44
<b>4 Webové rozhraní</b>	<b>47</b>
4.1 Využité technologie . . . . .	47
4.1.1 JavaScript . . . . .	47

4.1.2	PHP . . . . .	47
4.1.3	MySQL . . . . .	48
4.2	Architektura webové aplikace . . . . .	48
4.3	Správa chytrých elektroměrů . . . . .	49
4.3.1	Přidání nového zařízení . . . . .	49
4.3.2	Editace, odebrání zařízení . . . . .	49
4.4	Vizualizace výsledků měření . . . . .	50
4.5	Správa systému . . . . .	51
4.5.1	Uživatelské role . . . . .	51
4.5.2	Nastavení uživatelského účtu . . . . .	52
<b>Závěr</b>		<b>53</b>
<b>Literatura</b>		<b>55</b>
<b>Seznam symbolů, veličin a zkratk</b>		<b>57</b>
<b>Seznam příloh</b>		<b>59</b>
<b>A Obsah přiloženého CD</b>		<b>61</b>



# Seznam obrázků

1.1	Chytrá síť - provozovatelé rozvodných sítí . . . . .	22
1.2	Chytrá síť - firemní struktura . . . . .	22
1.3	Smart metering v ČR . . . . .	24
2.1	Referenční model ISO/OSI . . . . .	26
2.2	Komunikace pomocí service primitives . . . . .	30
2.3	Znázornění logických zařízení v rámci zařízení fyzického . . . . .	32
2.4	Umístění COSEM objektů v rámci logického zařízení . . . . .	33
3.1	Základní struktura DLMS koncentrátoru . . . . .	41
4.1	Přidání nového chytrého elektroměru do systému . . . . .	49
4.2	Sekce pro editaci/odebrání chytrého elektroměru . . . . .	50
4.3	Vizualizace výsledků měření . . . . .	51
4.4	Základní struktura DLMS koncentrátoru . . . . .	52
4.5	Editor uživatelského účtu . . . . .	52

# Seznam tabulek

1.1	Stav plnění EU doporučení . . . . .	23
2.1	Autentizace - seznam algoritmů . . . . .	31

## Seznam výpisů

3.1	Získání hodnoty napětí z chytrého elektroměru . . . . .	42
3.2	Metoda určená pro čtení atributů . . . . .	42
3.3	Obsah konfiguračního souboru “AppConfig.json” . . . . .	43
3.4	Obsah provozního XML souboru . . . . .	44
3.5	Struktura webového požadavku . . . . .	45

# Úvod

Internet umožnil vzniknout mnoha inteligentním systémům a chytrým zařízením, které mohou být vzájemně propojené a mají tu možnost spolu komunikovat. Jedním druhem těchto zařízení jsou i chytré měřiče tzv. „smart metery“. Chytré měřiče jsou schopné měřit nejrůznější veličiny. Může se jednat o měření průtoku vody, spotřeby plynu nebo elektrické energie.

Práce se zaměřuje na vytvoření zařízení, které má za úkol shromažďovat údaje z připojených chytrých elektroměrů, tzv. koncentrátor chytrých elektroměrů. Chytré elektroměry komunikují s koncentrátorem pomocí DLMS standardů přes TCP/IP protokol.

DLMS standardy vytváří a spravuje DLMS User Association[2]. Členové této neziskové asociace se snaží o sjednocení ekosystému chytrých elektronických měřičů po celém světě.

Data, která chytrý elektroměr poskytuje koncentrátoru tj. zařízení umožňující správu několika chytrých zařízení, jsou uloženy ve speciálních COSEM objektech. Tyto objekty jsou poté přenášeny pomocí média, nejčastěji pomocí Ethernetu nebo sériového spojení, mezi zařízeními.

V teoretické části budou rozebrány DLMS standardy – proces komunikace, specifikace služeb a bezpečnostní standardy. Dále zde budou uvedeny COSEM objekty umožňující přenos dat ve stanovených datových strukturách.

Nabyté znalosti z výše uvedených oblastí budou uplatněny při samotném zpracování praktické části práce. Ta se zaměřuje na popis principů a jednotlivých částí vytvořeného koncentrátoru. V neposlední řadě bude v této části předvedeno i grafické rozhraní koncentrátoru, které umožňuje uživateli správu koncentrátoru.

# 1 Úvod do chytrých měřicích zařízení

Technologie přispívají k rychlému rozvoji lidstva. Zlepšují kvalitu života díky precizním technologiím využívaných v lékařství. Usnadňují namáhavou fyzickou práci při stavebních činnostech budov či infrastruktur. Informují obyvatele o nejrůznějších událostech pomocí spletné datové sítě – internetu. To byl pouze zlomek toho, čeho nám technologie v dnešní době umožňují dosáhnout.

Právě internet je jedním z významných faktorů, který ovlivnil rychlost vývoje lidstva. Schopnost okamžitě přenášet myšlenky pomocí datových sítí umožnilo vzniku mnoha oblastí. Ať už se jedná o textová data ve formě webových stránek, textovou konverzaci na sociálních sítích nebo audiovizuální data v podobě online streamování videí, hudby nebo účasti se v audio a video konferencích.

Jednou z těchto oblastí, které dovolil vznik právě internet, je oblast chytrých měřicích zařízení tzv. Smart meterů. Smart metery umožňují odečty měřených veličin na dálku. Pomáhají nám lépe porozumět charakteru sítí a identifikovat zdroje zvýšených spotřeb.

Tato práce se zaměřuje především na chytré elektroměry. V oblasti smart meteringu jsou v dnešní době nejrozšířenější. Je to dáno tím, že elektřina, která je zapotřebí k napájení chytrých elektroměrů, je dostupná téměř na všech místech. Jediné, na co je poté potřeba myslet, je dostupnost internetu. Zřízení chytrých elektroměrů tedy vyžaduje určitou míru plánování.

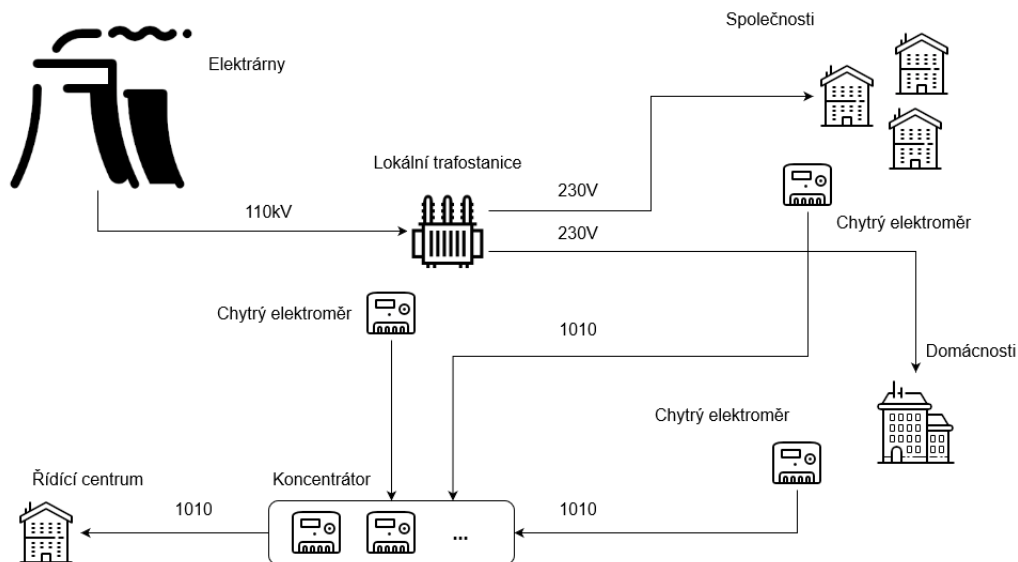
Chytré elektroměry mohou zřizovat jak provozovatelé rozvodných sítí[3], tak i jednotlivci. Provozovatelé rozvodných sítí zřizují koncentrátoři pro jednotlivé uzly a domácnosti - Obr. 1.1. To jim umožňuje lepší monitoring jednotlivých částí elektrických sítí. V případě jakýchkoliv problémů umožní provozovatelům chytrá síť rychleji reagovat na vzniklé krizové situace. Jednotlivci pak mají možnost vybavit chytrým elektroměrem své domácnosti a monitorovat spotřebu jednotlivých místností či spotřebičů. Společnostem umožňuje nasazením chytrých elektroměrů zefektivnit správu a řízení firmy Obr. 1.2.

Nevýhodami chytrých elektroměrů je především jejich vyšší cena, která brání širšímu a rychlejšímu rozšíření chytrých elektroměrů. Další nevýhodu lze spatřovat v nižší bezpečnosti, kdy je v případě špatné konfigurace chytrého elektroměru umožněn přístup útočníkům k naměřeným datům. Ti poté mohou zmapovat chod společností či domácností a s těmito nelegálně získanými daty dál nakládat.

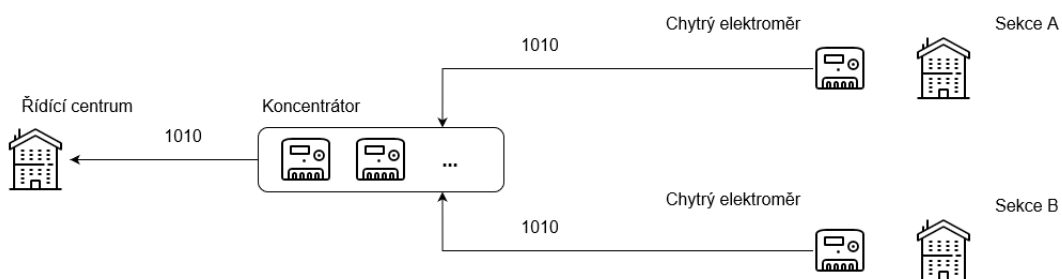
Chytré elektroměry měří spotřebu elektrické energie periodicky. Ve většině případech lze prodlevu mezi měřeními nastavit i manuálně. Frekvence měření se může pohybovat v rámci měsíců, týdnů ale i minut.

Pokud by se data z desítek chytrých zařízení sbírala na jedno místo, mělo by to za následek nezvladatelný datový provoz. Aby k tomuto nedocházelo, existují

koncentrátory. Úkolem koncentrátorů je především decentralizace sběrných míst tak, aby nedocházelo k zahlcení datových sítí. Dalšími hlavními úlohami koncentrátorů mohou být správa chytrých elektroměrů či vizualizace měřených výsledků.



Obr. 1.1: Chytrá síť u provozovatelů rozvodných sítí



Obr. 1.2: Chytrá síť ve firemní struktuře

## 1.1 Situace v Evropské unii

Evropská unie vydala v rámci řešení klimatické změny v roce 2009 tzv. „Třetí energetický balíček“, který klade členským státům EU nezávazný cíl – promoření chytrých elektroměrů ve výši 80% do roku 2020. „Pokud se zavádění inteligentních měřících přístrojů vyhodnotí pozitivně, musí být do roku 2020 inteligentními měřícími systémy vybaveno alespoň 80% spotřebitelů“ [4]. Tohoto cíle se však nepodařilo dosáhnout. Může za to kromě už vyjmenovaných nevýhod chytrých elektroměrů i povaha balíčku, která má formu pouze doporučení. Závisí tak pouze na jednotlivých zemích, zda-li budou ochotny tohoto cíle dosáhnout.

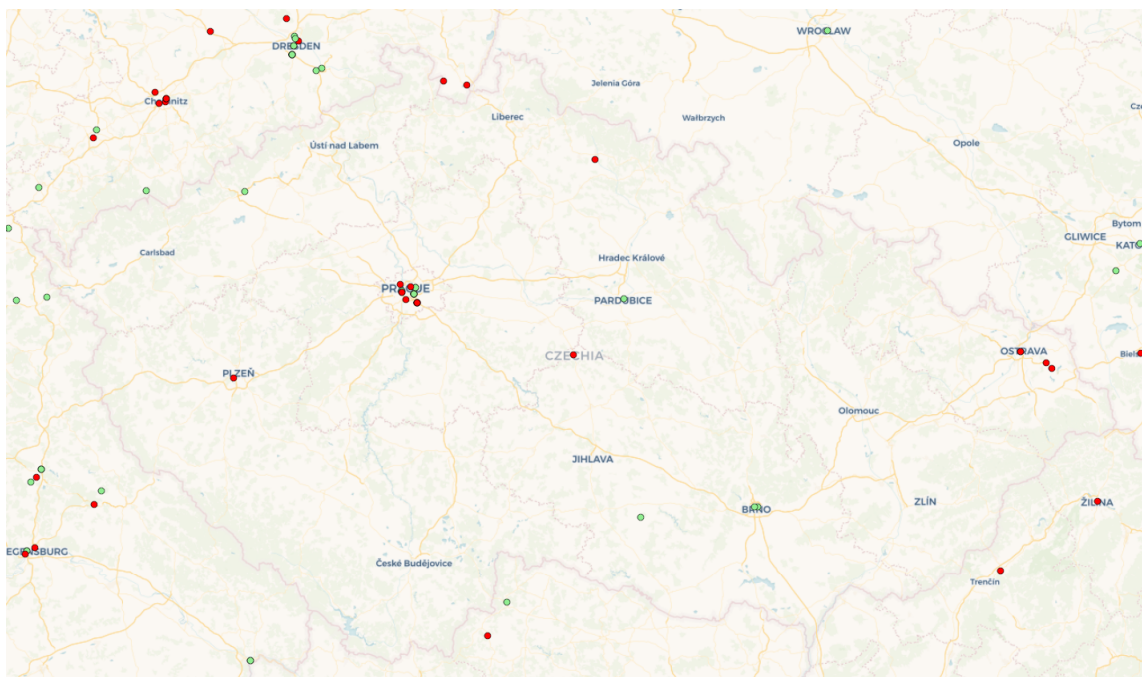
V Evropské unii se proto vytvořil roztržitý trh s chytrými sítěmi. Jak dosahují jednotlivé země daného plánu je uvedeno v Tab. 1.1.

Stav	Země
Splněno ( $\geq 80\%$ v roce 2020)	Itálie, Švédsko, Finsko, Nizozemsko
Dle plánu ( $\sim 80\%$ v roce 2020)	Francie, Španělsko, Řecko, Dánsko
Pozadu ( $< 80\%$ v roce 2020)	Velká Británie
Vlastní řešení	Německo, Belgie, Portugalsko

Tab. 1.1: Stav plnění „Třetího energetického balíčku“ v EU [1][5]

## 1.2 Situace v České republice

Česká republika v zavádění chytrých sítí zaostává za ostatními, vyspělými, zeměmi Evropské unie. Zatímco okolní země (Německo, Rakousko, Velká Británie) počítají počty projektů v řádů několika desítek až stovek, v České republice najdeme pouze jednotky desítek projektů, přičemž převážná část z nich jsou pilotní projekty nebo projekty vedené pod našimi vysokými školami. Veškeré aktivní projekty jsou zobrazeny na Obr. 1.3, kdy červené body značí pilotní projekty a zelené již zavedené projekty.



Obr. 1.3: Mapa projektů smart meteringu v ČR [6]



## 2 DLMS/COSEM

Device Language Message Specification a Companion Specification for Energy Metering, zkráceně DLMS/COSEM, specifikuje světově uznávané standardy pro výměnu dat mezi měřicími zařízeními. Jak už bylo zmíněno, standardy jsou vytvářeny a spravovány organizací DLMS User Association [2].

Ze specifikace DLMS/COSEM lze zjistit, že plní především tyto 3 hlavní funkce:

- **Modelování**

Zahrnuje ukládání dat do struktur, objektů a dále pravidla přístupu k těmto objektům pomocí unikátních identifikátorů. Datový model poté umožňuje pohled na elektroměrem podporované funkce.

- **Signalizace**

Komunikační služby a protokoly, které mapují jednotlivé prvky datového modelu na prvky aplikačního protokolu.

- **Transport**

Pokrývá veškeré protokoly a služby, které se podílejí na transportu zpráv mezi zařízeními.

Pro implementaci DLMS/COSEM se používají tzv. barevné příručky (anglicky Colored books), které spravuje právě výše zmíněná organizace DLMS User Association. Základní přehled a jednotlivé účely příruček jsou uvedeny níže:

- **Modrá příručka**

Anglicky Blue book, obsahuje popis tříd rozhraní COSEM a specifikaci modelu OBIS [7].

- **Žlutá příručka**

Anglicky Yellow book, popisuje metody a proces testování [8].

- **Zelená příručka**

Anglicky Green book, specifikuje DLMS/COSEM architekturu a využívané protokoly [10].

- **Bílá příručka**

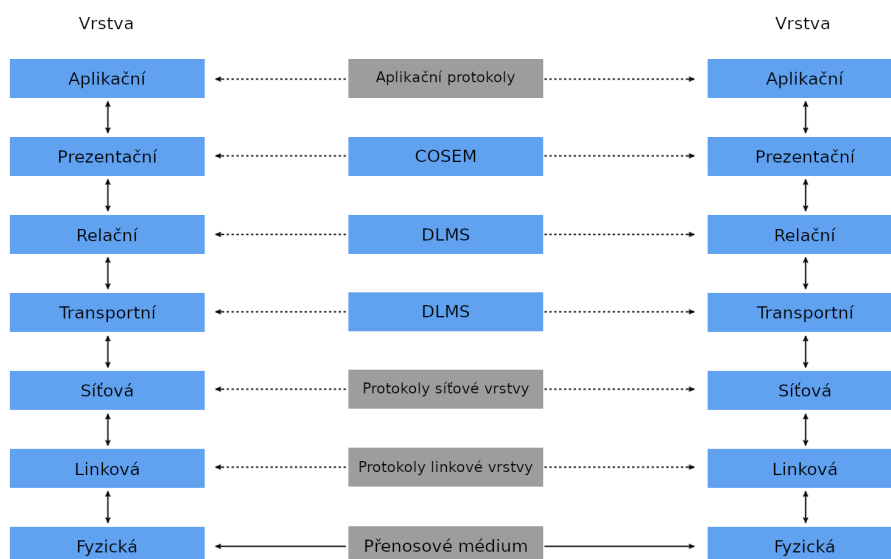
Anglicky White book, obsahuje seznam pojmů, kterých se používá v jednotlivých barevných příručkách.

DLMS/COSEM je založen na komunikačním modelu typu klient-server. Koncová zařízení – chytré elektroměry mohou plnit jak roli serveru tak i klienta. Většinou jsou však nakonfigurovány jako server. Serverová konfigurace umožňuje jejich snadnou konfiguraci přes síť. Zároveň se však chytrý elektroměr, který plní funkci serveru, musí nacházet ve stejné síti nebo mít statickou veřejnou IP adresu. Bez veřejné IP adresy se k serveru nelze připojit z jiných sítí.

## 2.1 DLMS/COSEM na modelu ISO/OSI

ISO/OSI komunikační model vypracovala organizace ISO s cílem standardizovat komunikaci v rámci počítačových sítí. Skládá z jednotlivých vrstev, které jsou na sobě nezávislé. Každá komunikační vrstva plní svůj vlastní úkol, přičemž může komunikovat pouze s přilehlými vrstvami.

DLMS/COSEM můžeme nalézt na třech vrstvách – prezentační, relační a transportní. COSEM objekty prezentují data ve speciálních strukturách a objektech. Z tohoto hlediska patří COSEM model na prezentační vrstvu. DLMS aplikační vrstvy specifikují služby, které poté přistupují ke COSEM objektům. Komunikační profily specifikují, jak se tyto služby mohou přenést přes různá média. Protokol DLMS proto existuje na relační a transportní vrstvě. Druh komunikace na nižších vrstvách není ve specifikaci DLMS/COSEM zahrnut a lze proto využít nejrozumnější způsoby komunikace – DLMS/COSEM je nezávislé na přenosovém médiu.



Obr. 2.1: Referenční model ISO/OSI

## 2.2 DLMS

Vznikl jako průmyslový komunikační protokol určený pro chytrá měřicí zařízení. Výměna dat mezi zařízeními je řešena pomocí modelu typu klient-server, kdy klient posílá požadavky na server, který na ně poté odpovídá.

### 2.2.1 Systémový název zařízení

Každé zařízení, které komunikuje pomocí DLMS/COSEM musí mít přiřazen unikátní identifikátor – adresu zařízení. Tato adresa zařízení by měla být statická a neměla by se měnit v čase.

Unikátní adresa zařízení se v DLMS označuje jako *Sys-T* [10]. Skládá se z 8 oktetů. První 3 oktety identifikují výrobce. Každý výrobce obdrží od DLMS User Association originální a jedinečný identifikátor, který se skládá pouze z velkých čísel. Ostatních 5 oktetů je definováno výrobcem tak, aby zaručil unikátnost — při tvorbě druhé části identifikátoru může výrobce vycházet z např. sériového čísla zařízení.

### 2.2.2 Proces komunikace

Komunikace pomocí DLMS zahrnuje především tyto tři kroky:

- **Inicializace spojení**

Navázání spojení s cílovým zařízením pomocí asociační služby AA, přičemž je poslán požadavek *A-Associate Request* do cílového zařízení, které nové spojení přijme nebo odmítne. Následně informuje klienta pomocí *A-Associate Response* odpovědi.

- **Komunikace**

Zahrnuje samotnou výměnu dat mezi zařízeními. V tomto kroku dochází ke čtením a zápisům dat na cílové zařízení viz. služby pro datové přenosy.

- **Uzavření spojení**

Ukončení spojení pomocí požadavku *A-Release Request* na cílové zařízení.

### 2.2.3 Specifikace služeb

Služby, které DLMS poskytuje můžeme rozdělit do 2 hlavních skupin.

První skupinou jsou služby zajišťující navázání a ukončení spojení tzv. asociační služby. Službu pro navázání spojení je třeba zavolat ještě před manipulací s daty (služby pro datové přenosy).

Druhou skupinou jsou pak služby pro datové přenosy, které se starají o samotný přenos dat mezi zařízeními. Pomocí služeb pro datové přenosy lze číst/zapisovat atributy na cílovém zařízení – chytrém elektroměru.

## 1. Asociační služby

- **AA navázání spojení**

Při AA připojování na cílové zařízení se používá aplikační PDU s názvem *A-Associate Request*. Cílové zařízení na tento požadavek odpovídá formou *A-Associate Response*.

- **AA ukončení spojení**

Při potřebě odpojení se odešle aplikační PDU s názvem *A-Release Request* cílovému zařízení. To odpoví aplikačním PDU – *A-Release Response*.

## 2. Služby pro datové přenosy

- **ShortName (SN) služby**

- **Služba Read**

Jedná o službu používanou ke čtení atributů z COSEM rozhraní. Lze vyčítat jednotlivé atributy zvlášť nebo hromadně. Případně lze pomocí této služby volat metody z COSEM tříd. Využívá aplikační PDU s názvy *readRequest* pro dotazy a *readResponse* pro odpovědi.

- **Služba Write**

Služba sloužící pro zápis jednoho a více atributů. Pomocí této služby lze zavolat i metody z COSEM tříd. Využívá aplikačních PDU s názvy *writeRequest* pro dotazy a *writeResponse* pro odpovědi.

- **Služba UnconfirmedWrite**

Služba téměř totožná se službou Write s tím rozdílem, že zapsání atributu není potvrzováno cílovým zařízením.

- **LongName (LN) služby**

- **Služba GET**

Jde o službu používanou ke čtení atributů z COSEM rozhraní. Lze vyčítat jednotlivé atributy zvlášť nebo hromadně. Využívá aplikační PDU s názvy *get-request* pro požadavek a *get-response* pro odpověď.

- **Služba SET**

Pomocí této služby se nastavují atributy COSEM objektu. Jsou použity aplikační PDU s názvy *set-request* a *set-response*.

- **Služba ACTION**

Služba, pomocí které se volají metody z COSEM rozhraní. Pro tuto službu se používají aplikační PDU s názvy *action-request* pro požadavky a *action-response* pro odpovědi.

### 2.2.4 Primitivní asociační služby

DLMS protokol má na aplikační vrstvě definované tzv. *service primitives*. Ty slouží k navázání spojení - asociaci, uzavření spojení a další operace. Proces komunikace pomocí těchto služeb je popsán níže:

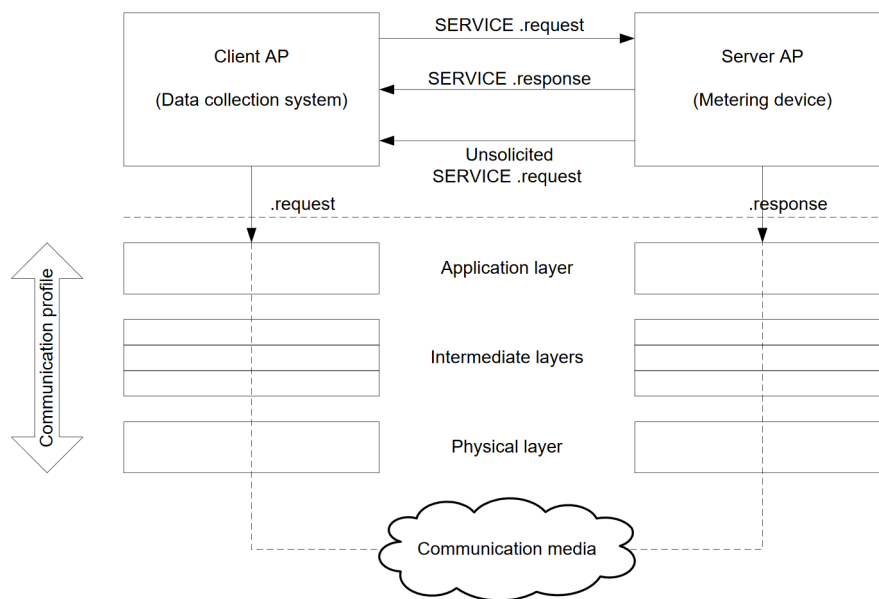
- **ASSOCIATION request**  
Klientská *service primitive*, žádající serverové zařízení o navázání spojení. Nižší vrstvy komunikačního modelu ISO/ISO požadavek zpracují a odešlou serveru.
- **ASSOCIATION indication**  
Serverová *service primitive*, oznamuje COSEM vrstvě přijetí žádosti na asociaci.
- **ASSOCIATION response**  
Serverová *service primitive*, odeslaná jako odpověď na ASSOCIATION request zpět klientovi.
- **RELEASE request**  
Klientská *service primitive*, odeslána v okamžiku kdy klient požaduje rozpojení asociace.
- **RELEASE indication**  
Serverová *service primitive*, oznamuje COSEM vrstvě přijetí žádosti o ukončení asociace.
- **RELEASE response**  
Serverová *service primitive*, odpověď na RELEASE request zpět klientovi.
- **RELEASE confirm**  
Klientská *service primitive* oznamující COSEM vrstvě ukončení spojení.

### 2.2.5 Bezpečnostní koncepty

Informační bezpečnost je v dnešní době velice důležitá. Chytrá zařízení komunikující pomocí DLMS/COSEM proto využívají několik technik, kterými se chrání před neautorizovanými a neautentizovanými přístupy. Mezi tyto techniky můžeme zařadit autentizaci, autorizaci a šifrování zpráv.

#### Šifrování zpráv aplikačních PDU

Pro šifrování zpráv aplikačních PDU se používá algoritmus AES v GCM módu. Jedná se o velice oblíbenou symetrickou, blokovou šifru s bloky o velikosti 128-bitů. Kryptografický algoritmus AES pracuje se třemi vstupními parametry. Prvním vstupním parametrem je zpráva, kterou je třeba zašifrovat. Druhým a třetím parametrem jsou pak klíč a inicializační vektor.



Obr. 2.2: Komunikace pomocí *service primitives* [10]

Na klíči se zařízení dohodnou ještě před samotným procesem šifrování. K tomuto účelu se nejčastěji používá kryptografický algoritmus Diffie-Hellman ve variantě eliptických křivek (ECDH). Diffie-Hellman algoritmus je založen na problému diskrétního logaritmu, díky kterému je pro útočníka téměř nereálné najít tajný klíč, na kterém se strany pomocí Diffie-Hellmanova algoritmu dohodly.

Inicializační vektor, zkráceně IV, slouží k tomu aby byly dvě obsahově stejné zprávy zašifrované odlišně. U chytrých zařízení je pro inicializační vektor použit systémový název zařízení *Sys-T*. Tím se zaručí odlišnost šifrovaných zpráv, zašifrovaných na odlišných zařízeních.

## Úrovně autentizace

DLMS/COSEM standard zavádí několik možných úrovní zabezpečení. Každá úroveň poskytuje určitou míru čtení a zápisu dat, případně manipulaci se zařízením. Pro přehled jsou tyto úrovně zabezpečení uvedeny níže:

- **No security authentication**

### Nezabezpečená úroveň

Jak už vyplývá z názvu úrovně, server nevyžaduje autentizaci. Čtení základních COSEM objektů z chytrého zařízení je umožněn všem.

- **Low Level Security authentication (LLS)**

**Nízká úroveň zabezpečení**

Chytré zařízení (server) při této úrovni vyžaduje heslo pro autentizaci klienta. Kontrola autentizace je prováděna již při samotném navazování spojení ve fázi AA připojování (COSEM-OPEN). Pokud bude heslo v pořádku, spojení je navázáno. V opačném případě je spojení odmítnuto serverem.

- **High Level Security authentication (HLS)**

**Vysoká úroveň zabezpečení**

Při této úrovni zabezpečení se navzájem prokazují obě strany. Využitý je pro tento proces model typu výzva-odpověď. Prokazování probíhá ve čtyřech krocích. V prvním kroku zašle klient výzvu serveru. Ve druhém kroku pak odpoví server s vlastní výzvou klientovi. Klient následně výzvu vyřeší a zašle odpověď serveru. Server nakonec vyřeší svou část výzvy a odpověď zašle zpět klientovi. Server i klient porovnají odpovědi protějškové strany s očekávaným výsledkem.

Pokud by byla odpověď správná, spojení se úspěšně vytvoří.

Algoritmus použitý při autentizaci definuje hodnota *mechanism\_id*. Tuto hodnotu definuje server při inicializaci spojení. Přehled možných hodnot *mechanism\_id* je uveden v Tab. 2.1.

Úroveň autentizace	mechanism_id
Bez zabezpečení	0
LLS – nízká úroveň zabezpečení	1
HLS – vysoká úroveň zabezpečení	2
HLS s použitým MD5 algoritmem	3
HLS s použitým SHA-1 algoritmem	4
HLS s použitým GMAC algoritmem	5
HLS s použitým SHA-256 algoritmem	6
HLS s použitým EC-DSA algoritmem	7

Tab. 2.1: Seznam algoritmů používaný při autentizaci [10]

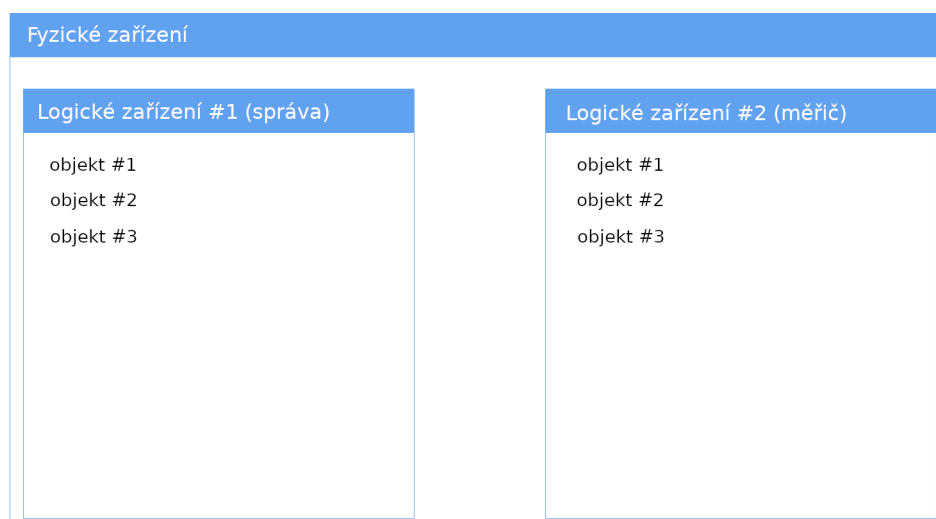
## 2.3 COSEM třídy rozhraní

COSEM umožňuje přístup skrze třídy rozhraní k objektům. Třídy rozhraní tedy slouží pro přístup a práci s daty, které jsou uloženy na cílovém zařízení např. na chytrém elektroměru.

Třídy rozhraní musí výrobci implementovat dle COSEM standardu. Veškeré povinné atributy a metody objektů musí být zachovány. Je poté na výrobci, zda-li přidá do objektů ještě nějaké atributy či metody navíc.

### 2.3.1 Struktura chytrého zařízení

Jedno fyzické zařízení může obsahovat několik logických zařízení. Logická zařízení pak slouží pro různé účely. Některá logická zařízení umožňují správu odpovídající části zařízení, jiná pak vystupují jako jednotlivé měřiče – lze z nich vyčítat hodnoty měření, případně provádět různé akce např. reset hodnot.

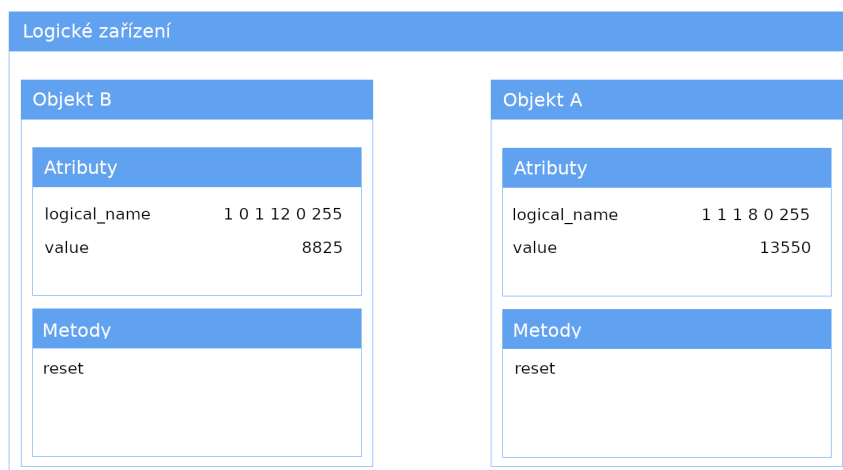


Obr. 2.3: Znázornění logických zařízení v rámci zařízení fyzického



Objekt obsahuje kolekci atributů a metody. Atributy uchovávají hodnoty v modelu typu klíč-hodnota. Prvním atributem v objektu je vždy *LongName* (LN). Následují i další atributy objektu např. měřené hodnoty. Metody objektu pak umožňují práci s atributy.

Objekty jsou vytvářeny ze tříd rozhraní. Současně také přebírají veškeré metody a atributy definované ve třídě rozhraní – třídy rozhraní jsou definovány COSEM standardem.



Obr. 2.4: Umístění COSEM objektů v rámci logického zařízení

### 2.3.2 Druhy přístupu ke COSEM objektům

Přístup ke COSEM objektům a atributům je zajištěn pomocí LN (LongName) a SN (ShortName) referencování. LN a SN si lze představit jako adresu, na které se objekty či atributy nachází.

- **LN reference**

Používá pro přístupu k celému objektu. Je-li žádoucí přistoupit k určité metodě či atributu, je potřeba uvést do parametrů žádosti také *class\_id* a *method/attribute\_index*.

- **SN reference**

Lze použít v situaci, kdy je potřeba přistoupit přímo k atributu nebo metodě bez zbytečného uvádění cesty. Tento zápis má výhodu v tom, že adresace je celkově jednodušší. SN referencování má však i nevýhodu – adresa může být mezi zařízeními odlišná.

### 2.3.3 Druhy COSEM tříd rozhraní

COSEM třídy lze rozdělit do několika skupin. Každá skupina reprezentuje určitou skupinu dat či nastavení dostupné na chytrém zařízení. Skupin existuje celá řada, pro přehlednost zde budou uvedeny pouze ty nejdůležitější [12]:

#### Parametry a data měření

- **Data**  
Třída s *class\_id=1* umožňující vytvořit triviální objekt se 2 atributy – LN a naměřenou hodnotou.
- **Register**  
Rozšiřuje třídu Data o měřenou jednotku.
- **Extended register**  
K registru přidává navíc atributy *status* – určovaný výrobcem a *capture\_time*, tedy čas, kdy došlo k zaznamenání hodnoty měření.
- **Demand register**  
Rozšiřuje třídu Extended register nejen o další atributy ale i funkcionality. Obsahuje údaje o aktuálních a předešlých průměrných hodnotách vypočtených z *n* posledních záznamů.
- **Activation register**  
Třída umožňující přepínání mezi tarify – denním a nočním režimem.

#### Řízení přístupu

- **Association SN**  
Třída umožňující připojenému klientovi získání všech SN objektů. V zařízení existuje pouze jeden objekt tohoto typu. Třidu lze najít na adrese SN = *0xFA00*. Třída obsahuje seznam všech tříd na zařízení – atribut *object\_list*. U každé třídy jsou uvedeny tyto informace:
  - *base\_name* – SN adresa,
  - *class\_id* – identifikační číslo rozhraní COSEM,
  - *version* – verze třídy,
  - *logical\_name* – LN adresa.
- **Association LN**  
Podobně jako Association SN obsahuje list všech tříd na zařízení – atribut *object\_list*. Navíc obsahuje i potřebná práva vyžadovaná pro přístup k určitým objektům.

## Čas, hranice událostí

- **Clock**

Třída uchovávající veškeré informace související s časem – časovou zónu, lokální čas, přesnost výpočtu času a další.

## 2.4 OBIS

COSEM Object Identification System, zkráceně OBIS, definuje unikátní identifikátory pro často používané datové položky měřené chytrými elektroměry. Používají se nejen k identifikaci naměřených hodnot, ale i k získávání informací o provozu zařízení.

### 2.4.1 OBIS třídy

OBIS unikátní identifikátor je ve výsledku šestice čísel tříd A-F, kde každé číslo má svůj význam a dohromady vytváří unikátní identifikátor s určitým významem. Unikátní identifikátory lze rozdělit do 6 tříd které jsou uvedeny níže. Každá třída umožňuje určitý set operací, který je závislý na předchozí třídě.

- **Třída A**

Identifikuje druh energie, se kterou zařízení pracuje. V případě elektřiny se jedná o hodnotu A=1.

- **Třída B**

Definuje číslo kanálu, ze kterého pochází datová komunikace chytrého elektroměru. Údaj je důležitý především pro datové koncentrátory a registrační jednotky chytrých zařízení.

- **Třída C**

Určuje druh měřených dat – proud, napětí, objem, teplotu. Povaha dat je závislá na třídě A. Ve specifikaci existují seznamy hodnot pro každou skupinu zařízení zvlášť – elektřina, voda, atd.

- **Třída D**

Definuje typy nebo výsledky zpracování dat ze třídy C na základě povahy dat ve třídě A. Příkladem může být minimum, maximum, průměrná hodnota a další.

- **Třída E**

Blíže určuje hodnoty měření specifikované třídami A-D.

- **Třída F**

Poslední třída může specifikovat účtovací období. Účtovací období, v závislosti na délce, obsahuje určitý počet naměřených hodnot.

## 2.4.2 OBIS identifikátor

Unikátní OBIS identifikátor je složen z jednotlivých tříd A-F. Jednotlivé třídy jsou zastoupeny hodnotami, nejčastěji od 0-255, přičemž každé číslo v dané kombinaci značí výběr položky dané specifikací. Unikátní OBIS identifikátor se zapisuje v následujícím formátu: 1-2:12.6.1\*125, kde význam jednotlivých hodnot je následující:

- 1-2:12.6.1\*125 – 1 značí, že se jedná o měření elektřiny,
- 1-2:12.6.1\*125 – 2 znamená, že data pochází z kanálu č. 2,
- 1-2:12.6.1\*125 – 12 měření napětí na jakékoliv fázi,
- 1-2:12.6.1\*125 – 6 určuje funkci – v tomto případě maximum,
- 1-2:12.6.1\*125 – 1 definuje, že jde o měření první harmonické frekvence,
- 1-2:12.6.1\*125 – 125 pak značí, že je vráceno pouze 25 posledních údajů.

### 3 Koncentrátor chytrých elektroměrů

Koncentrátor je hlavním jádrem této práce. Na starost má především sběr dat z určité množiny chytrých elektroměrů, které obdrží z uživatelského rozhraní – webového serveru.

Jelikož by měl koncentrátor pracovat na Raspberry PI, byl vytvořen s ohledem na multiplatformnost. Z tohoto důvodu byl při realizaci využit jazyk Java.

Koncentrátor sbírá naměřená data periodicky každou hodinu. Mezi data, která jsou sbírána, se řadí:

- **Síťové napětí**

Na vstupu chytrého elektroměru je měřeno síťové napětí, které je následně koncentrátorem sbíráno.

- **Síťová frekvence**

Koncentrátor shromažďuje informace o síťové frekvenci.

- **Vstupní zdánlivý výkon**

Jedná se o spotřebu elektrického proudu, kterou registrují provozovatelé síťové infrastruktury. Zahrnuje ztráty výkonu způsobené posunem periody střídavého proudu a další negativní jevy [17].

- **Výstupní reálný výkon**

Jedná se o výstupní hodnotu spotřeby energie, které připojené zařízení spotřebují. Hodnota je očištěna o většinu negativních jevů, zachycuje tedy reálnou spotřebu [17].

Veškerá data sbíraná koncentrátorem jsou ukládána v absolutních hodnotách. Naměřené hodnoty vstupního zdánlivého a výstupního reálného výkonu jsou také uloženy jako absolutní hodnoty z důvodu jednodušší implementace. Výpočet relativních hodnot je poté řešen za pomoci webové části této práce.

#### 3.1 Použité knihovny

Pro realizaci bylo potřeba využít knihoven třetích stran jelikož implementace DLM-S/COSEM specifikace by byla nad časový rámec této bakalářské práce.

První použitou knihovnou je open-source knihovna od společnosti GuruX. Tato společnost se zabývá činností okolo smart meteringu již od roku 1998. Vytváří a udržuje knihovny sloužící pro komunikaci s chytrými zařízeními. Jednotlivé knihovny jsou dostupné na jejich GitHubu [14]. K dispozici jsou knihovny pro programovací jazyky C#, Java a Python. Knihovny nabízejí nejrozličnější možnosti komunikace s chytrými zařízeními, zejména přes TCP/IP a sériové spojení.

Další použitou knihovnou sloužící pro serializaci a deserializaci JSON datových struktur je knihovna *json-simple*. Dostupná je na webových stránkách vývojářů [15].

## 3.2 Použitá zařízení

Koncentrátor chytrých elektroměrů lze provozovat na jakémkoliv zařízení s nejrůznějšími operačními systémy. Pro realizaci práce byl vybrán malý, jednodeskový počítač – Raspberry PI 3B.

Raspberry PI byl vyvinut britskou nadací Raspberry PI Foundation s cílem podpořit výuku informatiky na školách. První Raspberry PI nadace vydala již v roce 2012. V průběhu let se stalo Raspberry PI jedním z nejpopulárnějších jednodeskových počítačů na světě.

Oficiálně podporovaným operačním systémem, který je vyvíjen touto nadací, je Raspbian. Jedná se o jednu z linuxových distribucí založenou na Debianu. Tento open-source systém je dostupný pro procesory s architekturami ARM a x86.

Hardwarově na tom Raspberry PI 3B není vůbec špatně. Pohání jej 4-jádrový ARM procesor BCM2837 s frekvencí 1.2GHz. Operační paměť RAM má u tohoto modelu velikost 1GB, což bohatě stačí pro plynulý běh Raspbianu a méně výpočetně náročné aplikace. Základní deska disponuje 4x USB 2.0 porty, 100Mbit Ethernetem a HDMI portem. Kromě těchto základních rozhraní deska nabízí i CSI a DSI rozhraní určené pro kamery a displeje. Pro připojení externích zařízení a senzorů slouží 40-pinové GPIO rozhraní. Operační systém je uložený na MicroSD paměťové kartě [13].

## 3.3 Využití technologie

### 3.3.1 Java

Jedná se dnes o jeden z nejpoužívanějších jazyků na světě. Je to jazyk objektově orientovaný, kompilovaný do „bytecode“, univerzálního kódu, který je nezávislý na platformě. Multiplatformnost je poté zaručena pomocí JVM (Java Virtual Machine). JVM je vydávána pro různá systémová prostředí, ať už se jedná o Windows, Linux, Mac OS, Android nebo různé procesory (ARM, x86). JVM má za úkol převést „bytecode“ do instrukcí dostupných pro danou platformu a následně provést vykonání příkazů.

Díky své nezávislosti na platformě najde nejrůznější uplatnění. Javu dnes můžeme najít na stolních počítačích, kde multiplatformnost umožňuje psát aplikace pro různé operační systémy – v dnešní době nejpoužívanější Windows, Mac OS a Linux. Většina mobilních aplikací psaných pod systémem Android je vyvíjena pod Javou. Servery jsou dalším místem, kde se můžeme s Javou setkat. Dokonce i některé webové aplikace používají technologii JSP (JavaServer Pages).

## Historie Javy

Jazyk se původně, již od roku 1991, vyvíjel pod názvem Oak. Jméno získal po vedoucím vývojového týmu, který za tímto jazykem stál. Později však bylo zjištěno, že programovací jazyk s tímto jménem už existoval. Rozhodli se tedy pro název Java. V roce 1995 byla první beta verze oficiálně představená na veřejné konferenci [11]. V následujícím roce vydali plnou verzi Javy 1.0. Java se ve svém vývoji přizpůsobovala technologickým trendům. Kromě stolních počítačů se Java přizpůsobila i internetovému ekosystému. Na internetových stránkách ji bylo možné až do nedávna najít v podobě spustitelných doplňků. Z důvodů mnoha bezpečnostních incidentů ji však jednotliví výrobci webových prohlížečů přestali podporovat.

## 3.4 Odklon od specifikace DLMS/COSEM

Přestože jsou výrobci chytrých zařízení vedeni k implementaci všech částí dle specifikace DLMS/COSEM, někteří výrobci implementují části DLMS/COSEM dle vlastního uvážení. Tímto vzniká nekompatibilita mezi zařízeními od různých výrobců.

Právě takový problém vznikl se zařízeními, které byly k dispozici v laboratoři. Chytré elektroměry mají část DLMS/COSEM specifikace implementovanou dle výrobce.

Tato část práce byla značně zjednodušena díky bakalářské práci pana Bc. Davida Kohouta pod názvem “Zátěžový generátor zpráv DLMS/COSEM” [18], ve které popisuje odlišnosti implementace standardů výrobcem od specifikace DLMS/COSEM.

V následující části budou uvedeny důležité poznatky pana Bc. Davida Kohouta, které byly využity i při vypracování i této bakalářské práce.

### 3.4.1 Vytvoření spojení

Na začátku byly známy pouze IP adresy chytrých elektroměrů. Porty, na kterých naslouchají jednotlivá zařízení byly určeny dle seznamu registrovaných portů. Zařízení naslouchají na portech určené pro DLMS (4059).

Další parametr potřebný k navázání spojení je *clientID*. Pro zjištění této hodnoty byla vytvořena metoda testující jednotlivé *clientID* hodnoty. Parametr *clientID* lze taktéž zjistit z komunikace mezi chytrým elektroměrem a již existujícím koncentrátorem v laboratoři. Bylo zjištěno, že veškeré chytré elektroměry v laboratoři komunikují s *clientID*=17.

Další překážkou byla implementace mechanismu výzvy-odpovědi. Jelikož si výrobce tuto část implementoval dle svého uvážení, nepodařilo se pomocí knihovny,

která se připojuje pomocí standardizovaných procedur, vytvořit spojení. Bylo proto nutné daný systém obejít. Mechanismus výzvy odpovědi vždy generuje identickou výzvu – ta by se měla měnit v čase. Vzniká tímto náchylnost na tzv. *replay-attack*. Tato bezpečnostní zranitelnost byla využita pro získání odpovědi, která se poté při vytváření připojení odesílá na server.

### **3.4.2 Chybná implementace časové zóny**

Kvůli špatné implementaci časové zóny výrobcem nebylo možné přechíst data z profilových objektů. Tyto objekty obsahují především logy zařízení. Pokaždé, když bylo zapotřebí tyto objekty pomocí knihovny GuruX přechíst, knihovna hlásila problém ve spojitosti se špatným nastavením časové zóny. K vyřešení tohoto problému by bylo zapotřebí kontaktovat výrobce chytrého elektroměru, který by poté vydal opravnou verzi, ve kterém by chybnou implementaci časových zón opravil [16].



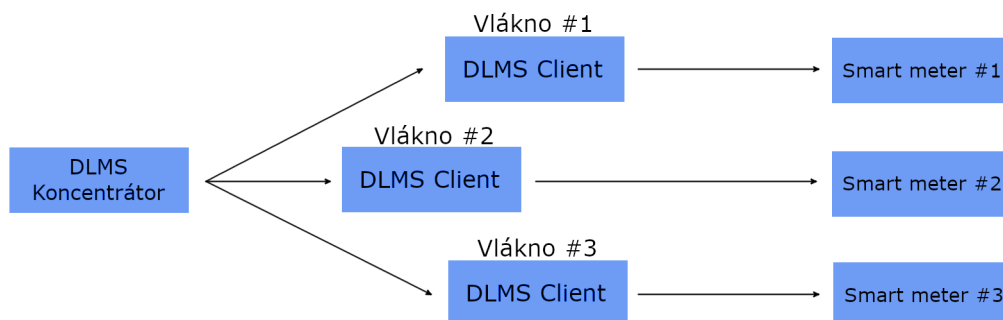
## 3.5 Funkční celky koncentrátoru

Koncentrátor se skládá z několika hlavních částí. Každá tato část má na starost určitou oblast funkcionality koncentrátoru. V této kapitole budou popsány jednotlivé části koncentrátoru.

### 3.5.1 Komunikace s chytrými zařízeními

Komunikace probíhá dle specifikací DLMS/COSEM pomocí TCP/IP protokolu. Ke komunikaci s chytrými zařízeními byla použita knihovna vyvinutá firmou GuruX.

Chytré elektroměry v laboratoři pracují v režimu serveru. Koncentrátor se poté k jednotlivým chytrým elektroměrům připojuje. Připojení k několika chytrým zařízením najednou není v jednovláknové aplikaci možné. Bylo proto nutné pro jednotlivé DLMS klienty vytvořit více aplikačních vlákn. Každé vlákno aplikace je nezávislé na hlavním vlákně. Hlavní vlákno se primárně používá pro přijímání a odesílání požadavků na webový server.



Obr. 3.1: Základní struktura DLMS koncentrátoru

### Čtení naměřených hodnot z chytrých elektroměrů

Pro čtení jednotlivých objektů je třeba definovat potřebné parametry. Koncentrátor používá pro přístup k atributům LN adresaci. Mezi potřebné parametry nutné pro přístup k atributu se řadí: LN adresa a Typ. Příprava parametrů pro čtení atributu je uvedena ve výpisu 3.1.

Výpis 3.1: Získání hodnoty napětí z chytrého elektroměru

```

1 private float ReadVoltage(Client client) {
2     SMObject smObject = new SMObject();
3     smObject.LnName = "1.0.32.7.0.255";
4     smObject.Description = "Ch._0_L1_Voltage_Inst._value";
5     smObject.Type = ObjectType.REGISTER;
6     smObject.AttributeCnt = 2;
7     return client.ReadObjectAttribute(smObject, 2);
8 }

```

Po definici parametrů nutných pro získání hodnoty atributu je třeba zavolat funkci starající se o získání dat z chytrého elektroměru. Tento účel plní metoda *ReadObjectAttribute()*.

Metoda nejprve načte asociační pohled, který obsahuje seznam všech existujících objektů na chytrém elektroměru. Následně je v tomto seznamu objektů vyhledán požadovaný objekt dle typu a LN adresy. Nakonec je zavolána metoda *read()* dostupná v knihovně GuruX, která zajistí přečtení vlastních dat.

Výpis 3.2: Metoda určená pro čtení atributů

```

1 public Object ReadObjectAttribute (SMObject smartMeterObject,
2                                     int attrIndex)
3 {
4     if(ConnectionInit()) {
5         LoadAssociationView();
6         GXDLMSObject object = clientReader.dlms.getObjects().
7                                     findByLN(smartMeterObject.Type,
8                                             smartMeterObject.LnName);
9         try {
10             Object obj = clientReader.read(object, attrIndex);
11             return obj;
12         }
13         catch (Exception e) {
14             e.printStackTrace();
15         }
16
17         ConnectionClose();
18     }
19     return null;
20 }

```

Po provedení všech operací je spojení s chytrým elektroměrem ukončeno.

### 3.5.2 Správa konfigurace

Konfigurace aplikace je ukládána do speciálního konfiguračního souboru. Bylo použito ukládání ve formě čitelného textu – JSON a XML. Konfigurační soubor lze nalézt v pracovním adresáři koncentrátoru s názvem “AppConfig.json”.

Provozní soubory jsou dynamicky vytvářeny za běhu programu. Obsahují asociční pohledy pro jednotlivé chytré elektroměry. Tyto soubory se nacházejí v adresáři */assoc\_views/*.

#### Konfigurační informace

V konfiguračním souboru “AppConfig.json” se nalézají veškeré informace nutné k úspěšnému připojení na chytrý elektroměr. Mezi tyto informace se řadí:

- IP adresy chytrých elektroměrů,
- hesla k chytrým elektroměrům,
- parametry *ClientID* k chytrým elektroměrům,

Následný výpis kódu zachycuje strukturu konfiguračních dat, uložené v konfiguračním souboru “AppConfig.json” (struktura konfiguračních dat byla kvůli zlepšení čitelnosti upravena).

Výpis 3.3: Obsah konfiguračního souboru “AppConfig.json”

```
1 {
2   "AppConfig":{
3     "ServerDiskLocation":"c:\\xampp\\htdocs\\",
4     "ServerAddress":"http://localhost2568\\",
5     "SmartMeters":[
6       {
7         "ClientID":17,
8         "ID":1,
9         "IPAddress":"10.183.94.105",
10        "Password":"cGFzcw=="
11      },
12      {
13        "ClientID":17,
14        "ID":2,
15        "IPAddress":"10.183.94.106",
16        "Password":"cGFzcw=="
17      }
18    ]
19  }
20 }
```

## Provozní informace

Provozní informace jsou ukládány do složky */assoc\_views/*. Složka obsahuje XML soubory s asociačními pohledy. Názvy XML souborů jsou voleny dle ID chytrých elektroměrů v systému.

Asociační pohledy, obsahující seznam všech dostupných objektů na chytrém elektroměru, jsou vytvořeny již při prvním úspěšném připojení koncentrátoru k chytrému elektroměru. Jelikož se tyto asociační pohledy nemění, je možné jejich opakované užití při pozdějších akcích.

Výpis 3.4: Obsah provozního XML souboru

```
1 <?xml version="1.0" ?>
2 <Objects>
3   <GXDLMSData>
4     <LN>0.0.0.0.1.255</LN>
5     <Description>Invalid</Description>
6   </GXDLMSData>
7   <GXDLMSData>
8     <LN>0.0.0.2.1.255</LN>
9     <Description>Ch. 0 Active firmware version</Description>
10  </GXDLMSData>
11  <GXDLMSClock>
12    <LN>0.0.1.0.0.255</LN>
13    <Description>Ch. 0 Clock object #1</Description>
14  </GXDLMSClock>
15 </Objects>
```

### 3.5.3 Komunikace s webovým serverem

Komunikace je zaručena pomocí třídy *WebCommunication.java*. Tato třída poskytuje jednoduché rozhraní pro posílání požadavků na webový server pomocí POST požadavků. Zároveň také monitoruje pomocí třídy *FileSystemWatcher* adresář, do kterého jsou zadávány příkazy, v podobě souborů webovým serverem.

Struktura požadavku je přesně definována modelem “WebCommunicationModel.java”. Model obsahuje jednotlivé parametry, které se přenáší pomocí POST požadavku na server. Struktura je znázorněna níže na ukázce zdrojového kódu.

Výpis 3.5: Struktura webového požadavku

```
1 public class WebCommunicationModel {  
2     public Action Action;  
3     public int MeterID;  
4     public String Data;  
5 }
```

Parametr *Action* obsahuje obecný typ požadavku, který může nabývat jedné z následujících hodnot:

- SmartMeterCreatedOK – přidání chytrého elektroměru skončilo úspěšně,
- SmartMeterCreatedFail – přidání chytrého elektroměru skončilo chybou,
- SmartMeterEditedOK – editace existujícího chytrého elektroměru skončilo úspěšně,
- SmartMeterEditedFail – editace existujícího chytrého elektroměru skončilo chybou,
- SmartMeterDeleteOK – odstranění existujícího chytrého elektroměru skončilo úspěšně,
- SmartMeterDeleteFail – odstranění existujícího chytrého elektroměru skončilo chybou,

Kromě již zmíněných akcí může parametr *Action* nabýt také hodnoty *SmartMeterMeasurement*. V tomto případě se společně s parametry *Action* a *MeterID* přenesou i parametr *Data* obsahující měřené hodnoty.

## 4 Webové rozhraní

Na Raspberry PI běží společně s koncentrátorem i webový server. Díky webovému serveru je umožněna správa chytrých elektroměrů připojených ke koncentrátoru. Webový server zároveň umožňuje vizualizace měřených hodnot ať už v podobě grafů či tabulek.

Další neméně důležitou oblastí je správa uživatelských účtů. Grafický systém umožňuje intuitivní přiřazování oprávnění a odstraňování uživatelských účtů.

Samozřejmostí je také správa uživatelského účtu – změna hesla nebo, pokud to bude zapotřebí, také smazání účtu.

Webový design stránek je zaměřen především na intuitivnost ovládání a responzivitou. Stránky podporují i mobilní zařízení – chytré telefony a tablety.

### 4.1 Využité technologie

#### 4.1.1 JavaScript

JavaScript je programovací jazyk umožňující provádět výpočty, měnit vlastnosti (vzhled, text) dokumentu nebo animovat různé prvky na webové stránce. Jedná se o klientský skript, což znamená, že stránky jsou staženy ze serveru a následně je kód, který je v nich obsažen spuštěn na počítači uživatele. Kód JavaScriptu je lehce čitelný ze zdrojového kódu zobrazené stránky, což je jeho výhoda i nevýhoda. Výhoda spočívá v tom, že každý začínající programátor se může začít učit JavaScript, který volně získá z prakticky každé stránky. Nevýhodu lze spatřovat v bezpečnosti aplikace, v případě, že je potřeba v kódu skrýt citlivý údaj.

#### 4.1.2 PHP

PHP (Hypertext Preprocessor), je serverový programovací jazyk. Spouští se na serveru ještě před tím, než jsou webové stránky staženy a spuštěny na straně klienta. Díky PHP lze vytvářet dynamické webové stránky, které se v čase mění.

PHP má výhodu ve své bezpečnosti, protože na rozdíl od JavaScriptu nelze z pohledu uživatele, který není správce serveru zdrojový kód aplikace nijak zobrazit. Velice důležitou vlastností PHP je schopnost navázat spojení s databázovým MySQL serverem, pomocí kterého lze ukládat, číst nebo mazat data z databázového serveru.

Kromě technologie PHP existuje řada dalších např. ASP, vyvíjená společností Microsoft, JSP, vyvíjená společností ORACLE a další.

### 4.1.3 MySQL

MySQL je nejrozšířenější, multiplatformní, databázový systém který může být provozován jak pod operačním systémem Windows, tak i pod Linuxem. Velice oblíbená kombinace v dnešní době: Linux + Apache (server, na kterém běží PHP) + MySQL. Příkazy jsou psány pomocí jazyka SQL, který je optimalizován především na rychlost. Správa MySQL serveru se nejčastěji řeší pomocí webové aplikace phpMyAdmin, ve které může uživatel spouštět příkazy, zobrazovat výsledky příkazů nebo je i optimalizovat.

## 4.2 Architektura webové aplikace

Webová aplikace je navržena za pomoci tzv. Třívrstvého aplikačního modelu. Cílem modelu je oddělení jednotlivých částí aplikace do logických celků. Každý logický celek má na starosti určitou činnost.

Aplikace je rozdělena na tyto 3 části:

- **DAL vrstva**

Vrstva přistupující k datům, anglicky Data Access Layer je vrstva sloužící pro komunikaci s databází. Nabízí jednoduché rozhraní pro práci s daty v databázi. Obsluhuje BL vrstvu.

- **BL vrstva**

Logická vrstva, anglicky Business Layer, je vrstva definující chování jednotlivých částí aplikace. Poskytuje metody, které využívá PL vrstva.

- **PL vrstva**

Prezentační vrstva, anglicky Presentation Layer. Vrstva, která má na starost vykreslování rozhraní a přijímání vstupních informací od uživatele.

## 4.3 Správa chytrých elektroměrů

Webový systém umožňuje základní správu chytrých zařízení. Chytrá zařízení lze pomocí webového rozhraní intuitivně přidávat, odebírat a upravovat.

### 4.3.1 Přidání nového zařízení

Ihned po přihlášení se uživateli zobrazí hlavní stránka se seznamem chytrých elektroměrů. Zde se nachází i tlačítko pro přidání nového chytrého zařízení. Po vyplnění požadovaných informací se nový chytrý elektroměr přidá k již existujícím v seznamu zařízení.

Při přidávání nového chytrého elektroměru není zadáván *ClientID* parametr. Hodnota tohoto parametru bude automaticky detekována koncentrátorem.

The screenshot shows the 'SmartMeters' web interface. At the top, there is a header with the title 'SmartMeters' and the subtitle 'Available smart meters'. On the right side of the header, there is a user profile icon and the text 'sample.email@vutbr.cz', 'Admin privileges', and links for 'System', 'Account', and 'Logout'. Below the header, there is a table with three columns: 'Name', 'Status', and 'IP Address'. The table contains one row with the data: 'Smart meter #1', 'OK', and '10.10.10.10'. Below the table, there is a form titled 'Add Smart Meter'. The form has three input fields: 'Name', 'IP', and 'Password'. At the bottom right of the form, there is a button labeled 'Add a new smart meter'.

Name	Status	IP Address
Smart meter #1	OK	10.10.10.10

Add Smart Meter

Name

IP

Password

Add a new smart meter

Obr. 4.1: Přidání nového chytrého elektroměru do systému

### 4.3.2 Editace, odebrání zařízení

Po výběru chytrého elektroměru je uživatel přesměrován na stránku umožňující prohlížet naměřené hodnoty a správu chytrého elektroměru. Část systému, která umožňuje editaci a odebrání chytrého zařízení je vyobrazena na Obr. 4.2.

Tato sekce není viditelná pro uživatele s úrovní oprávnění *Visitor*. Uživatelé, kteří disponují s oprávněním úrovně *Elevated* mohou měnit název chytrého elektroměru a konfiguraci připojení. Úroveň oprávnění *Admin* dále umožňuje vybraný chytrý elektroměr i odstranit.



Smart meter's settings

Change the name of the smart meter.

Smart meter #1

Change the name

Friendly name of the smart meter. Can contain special characters, diacritics, upper, lower case letters and digits.

Smart meter connection

10.10.10.10

.....

Save

After saving these parameters, the concentrator will check these credentials against the smart meter on a selected IP address.

Delete the smart meter.

Data for this smart meter will be deleted.

Delete

WARNING: THIS DECISION CAN'T BE TAKEN BACK.

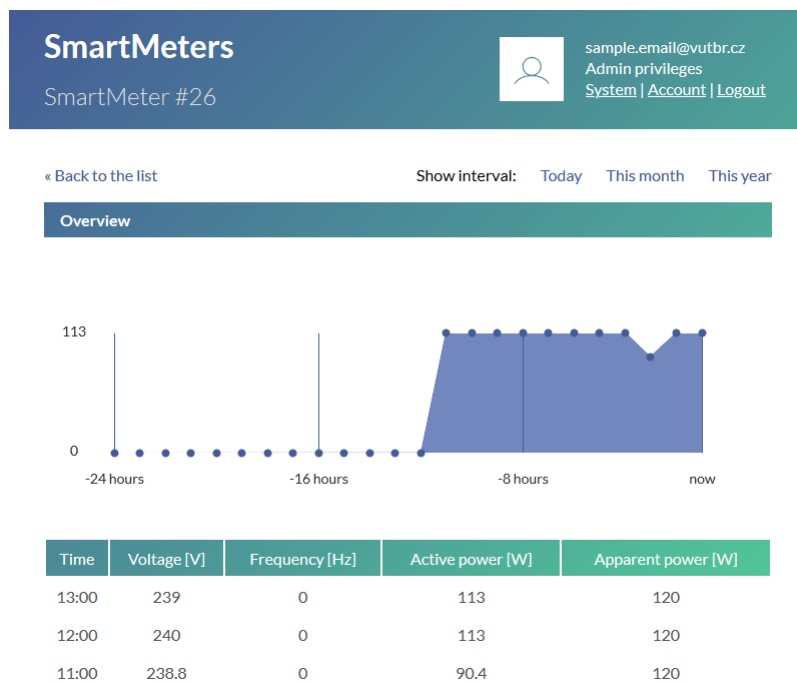
Obr. 4.2: Sekce pro editaci/odebrání chytrého elektroměru

## 4.4 Vizualizace výsledků měření

Vizualizace výsledků měření je v systému řešena pomocí grafů a tabulek. Mezi výhody grafů patří jejich přehlednost. Uživatel se rychle zorientuje v hodnotách, které vizualizují. Mezi nevýhody můžeme zařadit jejich strohost – malou informační hodnotu.

Ve velkých tabulkách je na druhou stranu těžké se zorientovat. Poskytují však vysokou informační hodnotu.

Webový systém proto kombinuje obě tyto metody vizualizace výsledků k dosažení požadovaného efektu – přehledný systém a zároveň i bohatý na informace. Výsledek kombinace obou těchto metod je znázorněn na Obr. 4.3.



Obr. 4.3: Vizualizace výsledků měření

## 4.5 Správa systému

Každý webový systém se skládá z uživatelů, kteří jej používají. Následující kapitoly jsou věnovány správě uživatelských účtů.

### 4.5.1 Uživatelské role

Uživatelé nabývají v systému určitých pravomocí. Každá úroveň pravomoci umožňuje provádět v systému určité povolené akce. Při registraci je každému uživateli přiřazena úroveň pravomoci *Visitor* – návštěvník.

Jednotlivé úrovně pravomocí jsou definovány následovně:

- **Visitor**

Umožňuje zobrazení seznamu chytrých elektroměrů, zobrazení výsledků měření. Je umožněna správa vlastního účtu – změna hesla a smazání účtu.

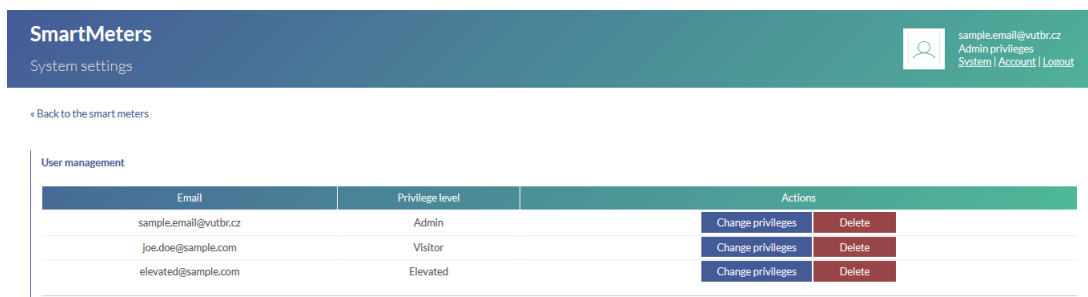
- **Elevated**

Obsahuje veškeré pravomoci úrovně *Visitor*. Navíc umožňuje změnu připojovacích parametrů k chytrému elektroměru a zobrazení logů.

- **Administrator**

Nejvyšší úroveň oprávnění umožňující navíc i vytváření a mazání chytrých elektroměrů ze systému.

V položce *system* na panelu profilu uživatele se nachází Editor oprávnění. Editor umožňuje změny úrovně oprávnění jednotlivých uživatelů. Platí přitom, že nejvyšší možnou úroveň oprávnění, kterou může uživatel udělit jinému uživateli se rovná úrovni oprávnění účtu provádějícího změny. Editor oprávnění je zobrazen na Obr. 4.4.



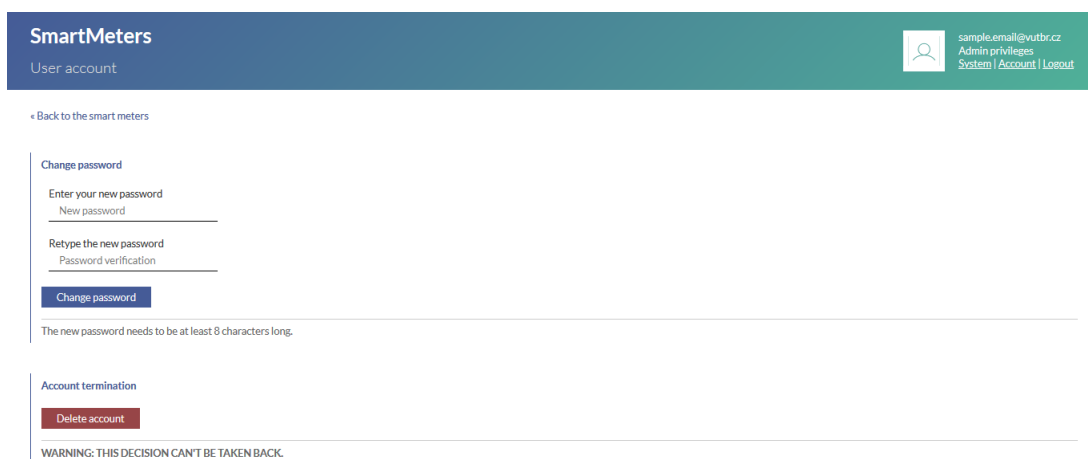
The screenshot shows the 'SmartMeters' system settings page. The header includes the system name, a user profile icon, and navigation links. The main content area is titled 'User management' and contains a table with three columns: 'Email', 'Privilege level', and 'Actions'. The table lists three users: 'sample.email@vutbr.cz' (Admin), 'joe.doe@sample.com' (Visitor), and 'elevated@sample.com' (Elevated). Each user has two actions: 'Change privileges' and 'Delete'.

Email	Privilege level	Actions
sample.email@vutbr.cz	Admin	<a href="#">Change privileges</a> <a href="#">Delete</a>
joe.doe@sample.com	Visitor	<a href="#">Change privileges</a> <a href="#">Delete</a>
elevated@sample.com	Elevated	<a href="#">Change privileges</a> <a href="#">Delete</a>

Obr. 4.4: Editor oprávnění

## 4.5.2 Nastavení uživatelského účtu

V systému je dostupná možnost správy vlastního účtu. K dispozici je možnost změny přístupového hesla či trvalého smazání účtu.



The screenshot shows the 'SmartMeters' user account page. The header includes the system name, a user profile icon, and navigation links. The main content area is titled 'User account' and contains two sections: 'Change password' and 'Account termination'. The 'Change password' section has two input fields for 'New password' and 'Password verification', a 'Change password' button, and a warning message: 'The new password needs to be at least 8 characters long.' The 'Account termination' section has a 'Delete account' button and a warning message: 'WARNING: THIS DECISION CAN'T BE TAKEN BACK.'

Obr. 4.5: Editor uživatelského účtu

## Závěr

V bakalářské práci byla analyzována situace v oblasti smart meteringu v Evropské unii i České republice. Byly nastíněny jednotlivé typy využití chytrých elektroměrů u různých typů subjektů, ať už se jedná o domácnosti či poskytovatele rozvodných elektrických sítí.

V teoretické části práce byla velká pozornost věnována komunikaci pomocí DLMS protokolu. Byly popsány jednotlivé kroky komunikace – od sestavení spojení, přenos dat až po ukončení spojení. Rozebrány byly i jednotlivé služby sloužící pro přístup k datům uložených na chytrých zařízeních. V neposlední řadě se teoretická část práce zaměřila i na COSEM objekty, jejich význam a účel.

Nabyté znalosti byly poté využity k realizaci koncentrátoru. Koncentrátor plní veškeré funkce dané zadáním, až na jednu. Nepodařilo se kvůli chybné implementaci časové zóny ze strany výrobce číst systémové logy na jednotlivých chytrých zařízeních. Vzhledem k povaze tohoto zásadního problému by náprava trvala příliš dlouho – pokud by tedy výrobce přistoupil na úpravu firmwaru zařízení. Koncentrátor bohužel může komunikovat jen se stejným druhem chytrého zařízení, umístěného v laboratoři. Důvodem je specifická implementace určitých částí chytrého elektroměru výrobcem.

Webový systém umožňuje intuitivní správu koncentrátoru. Má vyvinutý důmyslný systém správy uživatelských účtů.

# Literatura

- [1] SCULLY, Padraig. Smart Meter Market 2019: Global penetration reached 14% — North America, Europe ahead. *IOT ANALYTICS* [online]. Hamburg, c2020 [cit. 2020-05-15]. Dostupné z: <https://iot-analytics.com/smart-meter-market-2019-global-penetration-reached-14-percent/>
- [2] Who we are. *Dlms* [online]. Zug [cit. 2020-05-15]. Dostupné z: <https://www.dlms.com/the-association/who-we-are>
- [3] Inteligentní sítě vstupují do České Republiky. *ČEZ Distribuce* [online]. Děčín, c2020 [cit. 2020-05-15]. Dostupné z: <https://www.cezdistribuce.cz/cs/pro-media/tiskove-zpravy/153.html>
- [4] *SMĚRNICE EVROPSKÉHO PARLAMENTU A RADY 2009/72/ES*. In: . Bruselu, 2009, ročník 09, 2009/72/ES. Dostupné také z: <https://eur-lex.europa.eu/legal-content/CS/TXT/PDF/?uri=CELEX:32009L0072&from=EN>
- [5] GANGALE, Flavia, Julija VASILJEVSKA, Catalin Felix COVRIG, Anna MENGOLINI a Gianluca FULLI. *Smart grid projects outlook 2017*. Amsterdam: European Union, 2017. ISBN 978-92-79-68898-0. ISSN 1831-9424.
- [6] *Smart Grid Projects Map* [online]. Joint Research Centre, 2020 [cit. 2020-05-15]. Dostupné z: [https://ses.jrc.ec.europa.eu/sites/ses.jrc.ec.europa.eu/files/\u24/sgp\\_outlook\\_map/map.html](https://ses.jrc.ec.europa.eu/sites/ses.jrc.ec.europa.eu/files/\u24/sgp_outlook_map/map.html)
- [7] *Blue Book: COSEM Inteface Classes and OBIS Object Identification System*. DLMS [online]. Zug, Switzerland: DLMS User Association, 2017 [cit. 2020-05-15]. Dostupné z: <https://www.dlms.com/files/Blue-Book-Ed-122-Excerpt.pdf>
- [8] *Yellow Book: Conformance Testing Process*. DLMS [online]. Zug, Switzerland: DLMS User Association, 2017 [cit. 2020-05-15]. Dostupné z: <https://www.dlms.com/files/Yellow-Book-Ed-61-Excerpt.pdf>
- [9] *Green Book: Architecture and Protocols*. DLMS [online]. Zug, Switzerland: DLMS User Association, 2017 [cit. 2020-05-15]. Dostupné z: <https://www.dlms.com/files/Green-Book-Ed-83-Excerpt.pdf>
- [10] *Green Book: Architecture and Protocols*. DLMS [online]. Zug, Switzerland: DLMS User Association, 2017 [cit. 2020-05-15]. Dostupné z: <https://www.dlms.com/files/Green-Book-Ed-83-Excerpt.pdf>

- [11] *Úvod do Javy*. Brno, 2011. Dostupné také z: [https://moodle.sspbrno.cz/pluginfile.php/2917/mod\\_resource/content/0/java/zaklady/Java-uvod.pdf](https://moodle.sspbrno.cz/pluginfile.php/2917/mod_resource/content/0/java/zaklady/Java-uvod.pdf)
- [12] JIRKA, Matěj. *Framework DLMS/COSEM pro sběr datv AMM systémech*. Praha, 2016. Dilomová práce. České vysoké učení technické v PrazeFakulta elektrotechnická, Katedra řídicí techniky. Vedoucí práce Doc. Ing. Jiří Novák Ph.D.
- [13] *RaspberryPI* [online]. Cambridge: Raspberry Pi Foundation [cit. 2020-05-15]. Dostupné z: <https://www.raspberrypi.org/>
- [14] Gurux Ltd. *GitHub* [online]. San Francisco: GitHub, c2020 [cit. 2020-05-15]. Dostupné z: <https://github.com/gurux/>
- [15] *Json-simple* [online]. Davin Loegering [cit. 2020-05-15]. Dostupné z: <https://cliftonlabs.github.io/json-simple/>
- [16] Exception while reading 0.0.1.0.0.255 attribute 2. In: *GuruX* [online]. Tampere: Gurux, 1998 [cit. 2020-05-15]. Dostupné z: <https://www.gurux.fi/node/13660>
- [17] Fundamentals of Smart Metering: kWh and kVArh Meters. *Engineering Institute of Technology* [online]. Perth: Engineering Institute of Technology, c2019 [cit. 2020-05-15]. Dostupné z: <https://www.eit.edu.au/cms/resources/technical-resourses/fundamentals-of-smart-metering-kwh-and-kvarh-meters-2>
- [18] KOHOUT, David. *Zátěžový generátor zpráv DLMS/COSEM* [online]. Brno, 2019 [cit. 2020-05-15]. Dostupné z: [https://www.vutbr.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=193533](https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=193533). Bakalářská práce. Vysoké učení technické v Brně, Ústav telekomunikací, obor Informační bezpečnost.

## Seznam symbolů, veličin a zkratek

<b>AA</b>	Application Associations
<b>AES</b>	Advanced Encryption Standard
<b>ARM</b>	Advanced RISC Machine
<b>ASP</b>	Active Server Pages
<b>BL</b>	Business Logic
<b>COSEM</b>	Companion Specification for Energy Metering
<b>CSI</b>	Camera Serial Interface
<b>DAL</b>	Data Access Layer
<b>DLMS</b>	Device Language Message Specification
<b>DSI</b>	Display Serial Interface
<b>ECDH</b>	Elliptic-curve Diffie–Hellman
<b>GPIO</b>	General-purpose Input/Output
<b>HDMI</b>	High-Definition Multimedia Interface
<b>HLS</b>	High Level Security
<b>IV</b>	Initialization Vector
<b>JSON</b>	JavaScript Object Notation
<b>JSP</b>	JavaServer Pages
<b>JVM</b>	Java Virtual Machine
<b>LLS</b>	Low Level Security
<b>LN</b>	Long Name
<b>OBIS</b>	OBject Identification System
<b>PDU</b>	Protocol Data Unit
<b>PHP</b>	Hypertext Preprocessor
<b>PL</b>	Presentation Layer
<b>RAM</b>	Random-access Memory
<b>SN</b>	Short Name
<b>SQL</b>	Structured Query Language
<b>USB</b>	Universal Serial Bus
<b>XML</b>	Extensible Markup Language

# Seznam příloh

A Obsah přiloženého CD

61



## A Obsah přiloženého CD

/	kořenový adresář přiloženého CD
└─ README.txt	Instalační instrukce
└─ Koncentrator	Soubory koncentrátoru
└─ Koncentrator.zip	Zdrojové soubory
└─ Koncentrator.jar	Spustitelný program koncentrátoru
└─ WebovyServer	Soubory webového serveru
└─ WebovyServer.zip	Zdrojové soubory
└─ SmartMeters.sql	MySQL databáze
└─ ChytreElektromery	Soubory týkající se elektroměrů
└─ README.txt	Informace k adresáři
└─ SupportedObjects1.txt	Seznam objektů elektroměru #1
└─ SupportedObjects2.txt	Seznam objektů elektroměru #2
└─ SupportedObjects3.txt	Seznam objektů elektroměru #3
└─ Simulator_DLMS_koncentratoru-Bus.pdf	Elektronická verze práce